# Audit Report: Perfectly formed web development team - gotoAndPlay

**Website:** https://play.ee/
**Date:** 23.10.2026
**Audit Coverage:** 100% — all sources returned data
**Confidence:** high

**Pages Audited (1 of 1):**

- https://play.ee/

## Summary of results

**Overall Score:** 87 / 100
**Status:**  Needs Improvement

A strong, well-built site: PSI mobile 93 / desktop 100, LCP 2.6 s mobile (0.6 s desktop), CLS 0.000, TBT 1 ms and TTFB 44 ms — Core Web Vitals are effectively clean, and the only performance drag is Speed Index 4.11 s mobile (599 ms desktop) driven by render-blocking resources (est. 1,290 ms savings). Accessibility is close to passing (Lighthouse 96/95, axe 0 violations) but Lighthouse's color-contrast audit scores 0.00 at weight 7 and axe left 37 contrast nodes undetermined, so contrast is a real AA gap. SEO sits at 92 on both strategies, held back by 9 non-descriptive "read more" links, failing tap-target and structured-data audits. Security basics PASS (HSTS, nosniff, framing, TLS, no mixed content, WordPress hardening all good), but the CSP is frame-ancestors-only with no default-src/object-src, 2 of 2 cross-origin jsDelivr scripts lack SRI, and HSTS lacks preload — advanced-tier gaps, not baseline failures. W3C reports only 4 errors (WP Rocket's `defer` on `type="text/rocketlazyloadscript"` data blocks) and the page is light at 186 KB with 50 SVG images, so image assets are essentially a non-issue.

### Per-page scores

 Needs Improvement · https://play.ee/

| Score | Performance | Accessibility | Best Practices | SEO | Security |
|---|---|---|---|---|---|
| 87 | 93 | 96 | 100 | 92 | PASS |

## PageSpeed Insights — Mobile vs Desktop

*Lower is worse for Performance; higher is worse for LCP and CLS. Worse value is **bolded**.*

| Strategy | Performance (M / D) | LCP (M / D) | CLS (M / D) |
|---|---|---|---|
| Mobile vs Desktop | **93** / 100 | **2.58 s** / 638 ms | 0.000 / **0.007** |

## Optimization Checklist

**3 of 3 passing** — 3 pass · 0 warn · 0 fail · 5 n/a

| Item | Status | Detail |
|---|---|---|
| Page caching plugin / CDN active | **Pass** | Caching plugin detected (WP Rocket) |
| Response compressed (gzip / brotli) | **Pass** | Document response is compressed with gzip. |
| Images lazy-loaded | N/A | No raster  elements found (50 SVGs excluded). |
| Hero image eagerly loaded | N/A | No raster  elements found (50 SVGs excluded). |
| Hero is a real  (not a CSS background-image) | N/A | No CSS background-images detected on raster-image-eligible elements. |
| Responsive images (srcset / ) | N/A | Only 0 raster images on the page (50 SVGs excluded) — responsive-image rule does not apply. |
| Reasonable number of image sizes | N/A | Too few raster images to evaluate srcset width variety. |
| JS scripts not blocking in | **Pass** | No render-blocking scripts in . |

## Fixes

### Priority 1: Critical

*Immediate action — impacts user experience, search rankings, or site safety.*

**1A. Fix insufficient color contrast** `Accessibility`

- **Impact:** WCAG 1.4.3, Lighthouse accessibility score
- **Problem:** Lighthouse's `color-contrast` audit scores 0.00 (weight 7) on mobile, and axe-core could not determine contrast for 37 nodes (alpha/gradient backgrounds), so the failure is confirmed but not fully enumerated.
- **Solution:**
  - Identify the failing text/background pairs from the Lighthouse color-contrast details panel and raise them to ≥4.5:1 (≥3:1 for large text).
  - For the 37 axe-incomplete nodes, check computed colors including alpha — semi-transparent overlays over images are the usual culprit.
  - Prefer solid backgrounds or a darker text token, e.g. `color: #4a4a4a` on `#fff` instead of a light grey.

### Priority 2: Important

*Essential for compliance, user reach, and search visibility.*

**2A. Eliminate render-blocking resources to cut Speed Index** `Performance`

- **Impact:** Speed Index, FCP, LCP
- **Problem:** `render-blocking-insight` is flagged high with estimated savings of 1,290 ms; mobile Speed Index is 4.11 s versus 599 ms on desktop, and FCP is 1.99 s mobile vs 514 ms desktop.
- **Solution:**
  - Inline critical above-the-fold CSS and load the single external stylesheet (34.3 KB) asynchronously.
  - Defer or self-host the two `fast.fonts.net` requests (2 requests, 0 bytes returned) — use `font-display: swap` and `<link rel="preload" as="font">` for the fonts actually used above the fold.
  - Review the 24 scripts: 1 is still render-blocking; move it to `defer` or load it after first paint.
  - Re-measure Speed Index after the change — the 3.5 s mobile/desktop gap points at throttled CSS/JS delivery, not server latency (TTFB is 44 ms).

**2B. Replace vague repeated link text with descriptive labels** `SEO`

- **Impact:** SEO (link-text audit), screen-reader navigation
- **Problem:** `link-text` fails with 9 links lacking descriptive text; the HTML inventory shows "read more" repeated ×9, plus "view our instagram feed" ×3, "visit our facebook page" ×3 and "visit our linkedin page" ×3.
- **Solution:**

- Give each "read more" a unique, descriptive label, e.g. `<a href="/work/acme">Read the Acme case study</a>`.
- Where the visible text must stay short, add an accessible name: `<a href="…" aria-label="Read more about the Acme case study">read more</a>`.
- Social links already carry context in their text; keep them but ensure each is distinguishable (e.g. "gotoAndPlay on Instagram").

**2C. Add Subresource Integrity to cross-origin scripts** `Security`

- **Impact:** Supply-chain integrity (jsDelivr compromise)
- **Problem:** 2 of 2 cross-origin scripts lack an `integrity` attribute — both `cookieconsent.umd.js` and `iframemanager.js` load from `cdn.jsdelivr.net` unpinned.
- **Solution:** Pin exact versions and add SRI hashes:

```
<script
src="https://cdn.jsdelivr.net/gh/orestbida/cookieconsent@3.0.1/dist
/cookieconsent.umd.js"
        integrity="sha384-…" crossorigin="anonymous" defer>
</script>
```

  Generate the hash with `openssl dgst -sha384 -binary file.js | openssl base64 -A`. Better still, self-host both libraries — they are small (10.1 KB and 3.0 KB).

**2D. Fix tap targets and add valid structured data** `SEO`

- **Impact:** SEO score (92), mobile usability, rich results
- **Problem:** PSI's modeled SEO audits `tapTargets` and `structuredData` both fail; the single JSON-LD block is typed "Unknown" rather than a recognized schema.org type.
- **Solution:**
  - Enlarge interactive targets to at least 44×44 px with adequate spacing (WCAG 2.5.8) — check nav links, social icons and the language switcher.
  - Replace the unrecognized JSON-LD with a valid `Organization` (or `ProfessionalService`) graph including `name`, `url`, `logo`, `sameAs` and `contactPoint`.
  - Validate with the Rich Results Test after the change.

### Priority 3: Best Practice

*Recommended for long-term maintainability.*

**3A. Harden the Content-Security-Policy** `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only `frame-ancestors 'self';` — `default-src` is missing and `object-src` is not `'none'`, so no script or resource restrictions apply. Site signals show no auth, payments, user content or commerce, so this is a low-urgency hardening item rather than a critical gap.
- **Solution:** Add the cheap, low-risk directives now and move to a nonce-based policy if the site ever adds forms or user content:

  ```
  Content-Security-Policy: default-src 'self'; object-src 'none';
  base-uri 'self'; frame-ancestors 'self';
  ```

  When you are ready for script restrictions, use nonce + `'strict-dynamic'` rather than a flat allowlist.

**3B. Complete advanced header hygiene (HSTS preload, COOP/CORP, server banner, login path)** `Security`

- **Impact:** Transport hardening, origin isolation, attack-surface reduction
- **Problem:** HSTS lacks the `preload` directive, COOP and CORP are missing, `Server: Apache` discloses the stack, and the WordPress login is served on the default `/wp-login.php` path.
- **Solution:**

  ```
  Header always set Strict-Transport-Security "max-age=63072000;
  includeSubDomains; preload"
  Header always set Cross-Origin-Opener-Policy "same-origin"
  Header always set Cross-Origin-Resource-Policy "same-origin"
  Header unset Server
  ```

  - Submit the domain at hstspreload.org once the header is live.
  - Move the login form behind a non-default path or add IP/2FA restrictions.

**3C. Fix the 4 W3C validation errors from WP Rocket lazyload** `Best Practices`

- **Impact:** HTML validity, script execution semantics
- **Problem:** W3C reports 4 errors, all the same: a `<script type="text/rocketlazyloadscript">` data block carries a `defer` attribute, which is invalid for non-JavaScript MIME types (first at line 2064).
- **Solution:**

  - Update WP Rocket to a version that does not emit `defer` on `text/rocketlazyloadscript` blocks, or exclude the affected scripts from

delay-JS execution.

  - Alternatively, switch the delayed scripts to `type="text/javascript"` with `defer` and let the lazyloader rewrite them at runtime.

**3D. Trim unused jQuery and tighten document caching** `Performance`

- **Impact:** Unused JS, repeat-visit caching
- **Problem:** `jquery.bfe1bb19d13b3c17b682.min.js` wastes 23 KB and produces a 52 ms long task; the document is served with `cache-control: max-age=0` despite WP Rocket being active.
- **Solution:**
  - Audit whether jQuery is still required by theme/plugin code; if only a few plugins need it, replace with vanilla equivalents or load it only on pages that use it.
  - Configure WP Rocket to serve cached HTML with a positive `max-age` (e.g. `max-age=600, s-maxage=3600`) so repeat visits avoid revalidation.
  - Add `loading="lazy"` to the one image missing explicit `width` / `height` to avoid future CLS regressions.